Access Grid 2.0

Developer Overview

Retreat 2003

Overview

- Requirements Summary
- Architectural Overview
- Design Details
- Developer Entry Points

Requirements for AG 2.0

- Provide the functionality of AG 1.0
- Support for plug-in applications
- Tightly integrate the core functionality
- Improve Usability
- Wider Range of Client Platforms
- Integrate Grid Computing

AG 1.0 Functionality

- Basic Media Communications
  - Audio, 16KHz 16 bit, uncompressed
  - Video, 352x288, 25fps, H.261
  - Text, via MOO
- Venue Navigation

Support for “plug-in” Applications

- Two ways to collaborate:
  - Application Services
  - Shared Applications
- Application Services are third party applications, found in Venues
- Shared Applications are Venue Client “plug-ins” – that provide collaborative functionality
- These both rely on interfaces that are exposed in the Venue and Venue Client

Tightly Integrate Core Functionality

- Network Connectivity
  - Indication of state
  - Fallback to unicast
- Consistent Shared View of the Venue
- Text Communication
- Data Sharing

Improve Usability

- Users operate personal nodes themselves
- Users can extend this experience to operate shared nodes
- Clear, Unified User Interface
- Clear indication of problems
- Integrated Network Failure Recovery

Wider Range of Client Platforms

1. Advanced Node – Tiled Display, Multiple Video Streams, Localized Audio
2. Room Node – Shared Display, Multiple Video Streams, Single Audio Stream (AG 1.x Node)
3. Desktop Node – Desktop Monitor, Multiple Video Streams, Single Audio Stream (AG 1.X PIG)
4. Laptop Node – Laptop Display, Single Video Stream, Single Audio Stream
5. Minimal Node – Compact Display, Single Video Stream, Single Audio Stream

Integrate Grid Computing

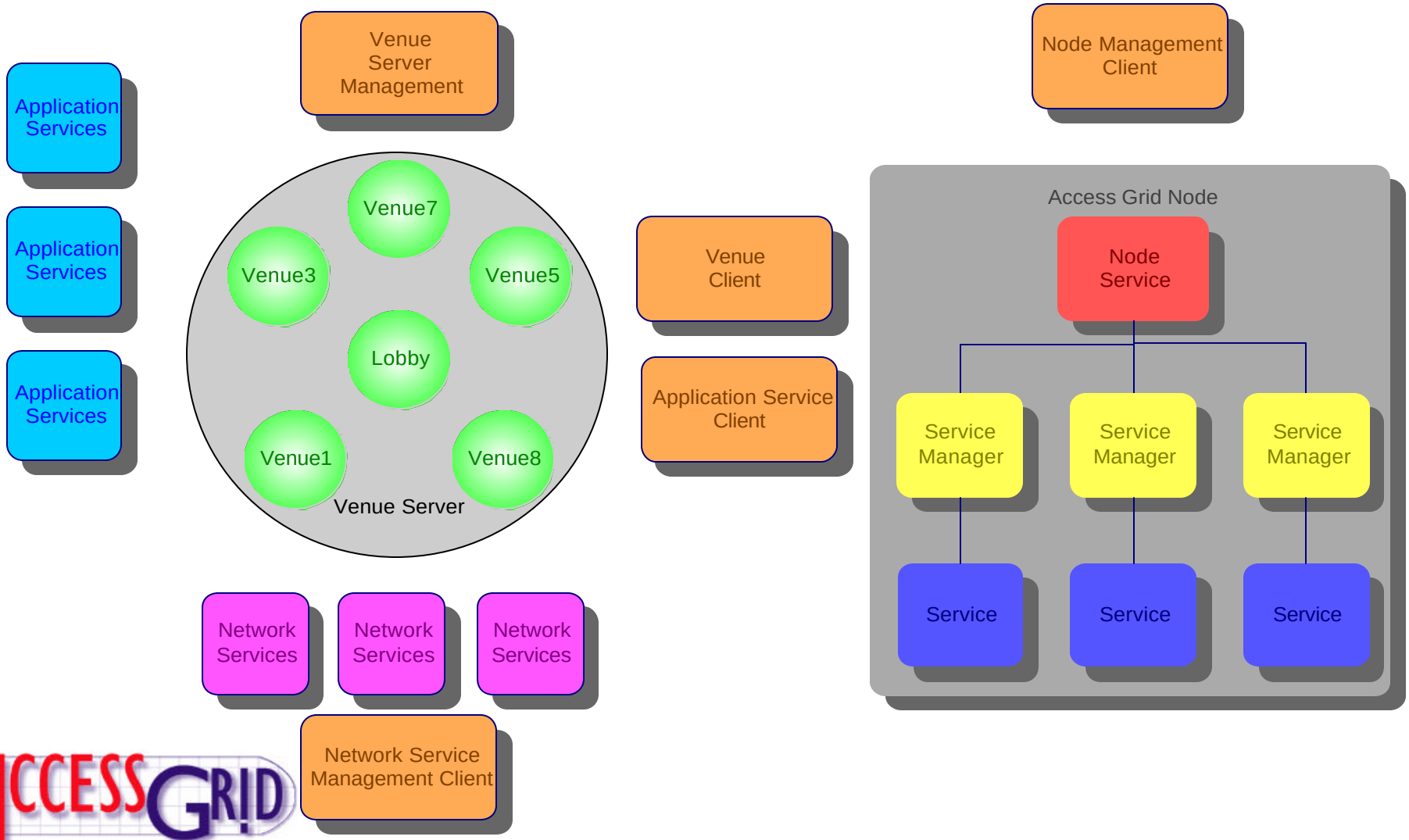
A small thought experiment to show where we're going:

- Imagine being able to gather your collaborators together in a Virtual Venue.
- From your Venue Client you can:
  - Browse and select an initial data set
  - Select an application you'd like to run with the data
  - Select a computing resource that can execute the application
  - Specify that you'd like to have a shared visualization of the resulting data when it appears in the Virtual Venue

Integrate Grid Computing (cont'd)

- The first steps are happening under the covers now
- Security is being done with GSI 2.0
- All data transfer is being done via Globus IO
- All sockets are secured with Globus IO

Architectural Overview



ACCESS GRID

Retreat 2003

Virtual Venues

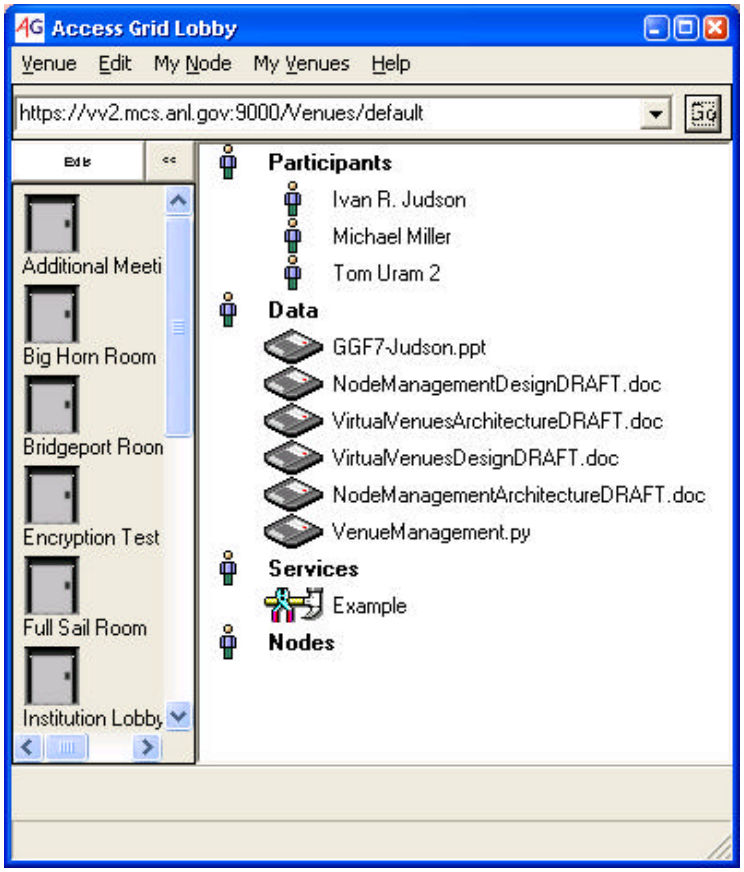
- What is a Virtual Venue?
  - A Virtual Venue is a virtual space for people to collaborate
- What do Virtual Venues provide?
  - Entry/Exit Authorization Information
  - Connections to other Venues
  - Coherence among Users
    - Venue Environment, Users, Data
  - Client Capabilities Negotiation
    - List of Available Network Services
    - Keep track of resulting Stream Configurations
  - Applications
- Virtual Venues have two interfaces
  - Administrative – Venue Management Software
  - Client – Virtual Venue Client Software

Network Services

- Network Services
  - Provide a middleware layer for enabling the richest collaborations
  - Are invisible to Venues Clients, used by Virtual Venues
  - Primarily Transform streaming data
  - Can be anywhere on the network
  - Can be composed to build complex solutions:
    - Venue Audio Stream → Audio Transcoder → Audio to Text → Two-Way Pager
    - Two-Way Pager → Text to Audio → Audio Transcoder → Venue Audio Stream

- ANL is working on Network Services for
  - Audio Transcoding (16KHz ? 8KHz)
  - Video Stream Selection

Virtual Venues Client



- Enable face-to-face meeting activities
- What can be done:
  - Sharing Data
  - Shared Applications
- Applications:
  - Distributed PowerPoint
  - Shared Web browser
  - Whiteboard
  - Voting Tool
  - Question & Answer Tool
  - Shared Desktop Tool
- Integrate legacy single-user apps

ACCESS GRID

Retreat 2003

Access Grid Nodes

- Access Grid Nodes
  - Comprise a set of collaboration resources
  - Expose those resources through Node Services
- Basic Node Services include:
  - Audio & Video Services
  - Network Performance Monitoring Service
  - Network Reliability/Fallback Service
  - Leashing Service – Registering presence with a shared node
- Extended Node Services could be:
  - Display Service with enhanced layout control
  - Video Service supporting new CODECs
  - Automatic performance adaptation
  - Application Hosting Service

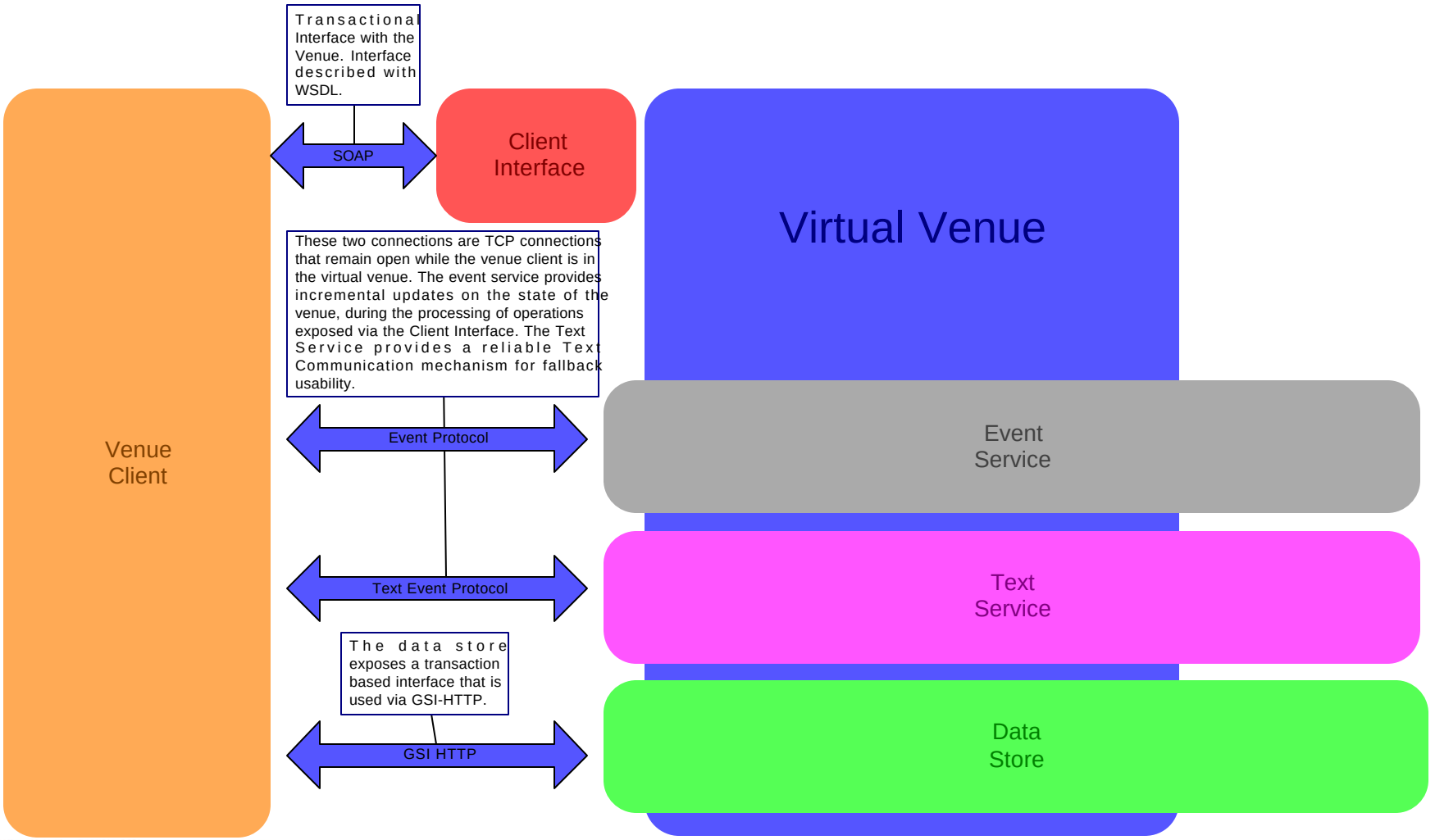
Application Services

- Application Services
  - Expose an external services to Virtual Venues
  - Provide the Service with the same coherent information as a Venue Client.
    - Authorization Information – list of users
    - Service Availability – list of services
    - Data Availability – list of data
  - Example Application Services:
    - Voyager – provides archive and playback services
    - Scheduling – provides scheduling services for Venues and Nodes

Design Details

- Virtual Venues
- Virtual Venues Client
- Network Services
- Node Services
- Application Services

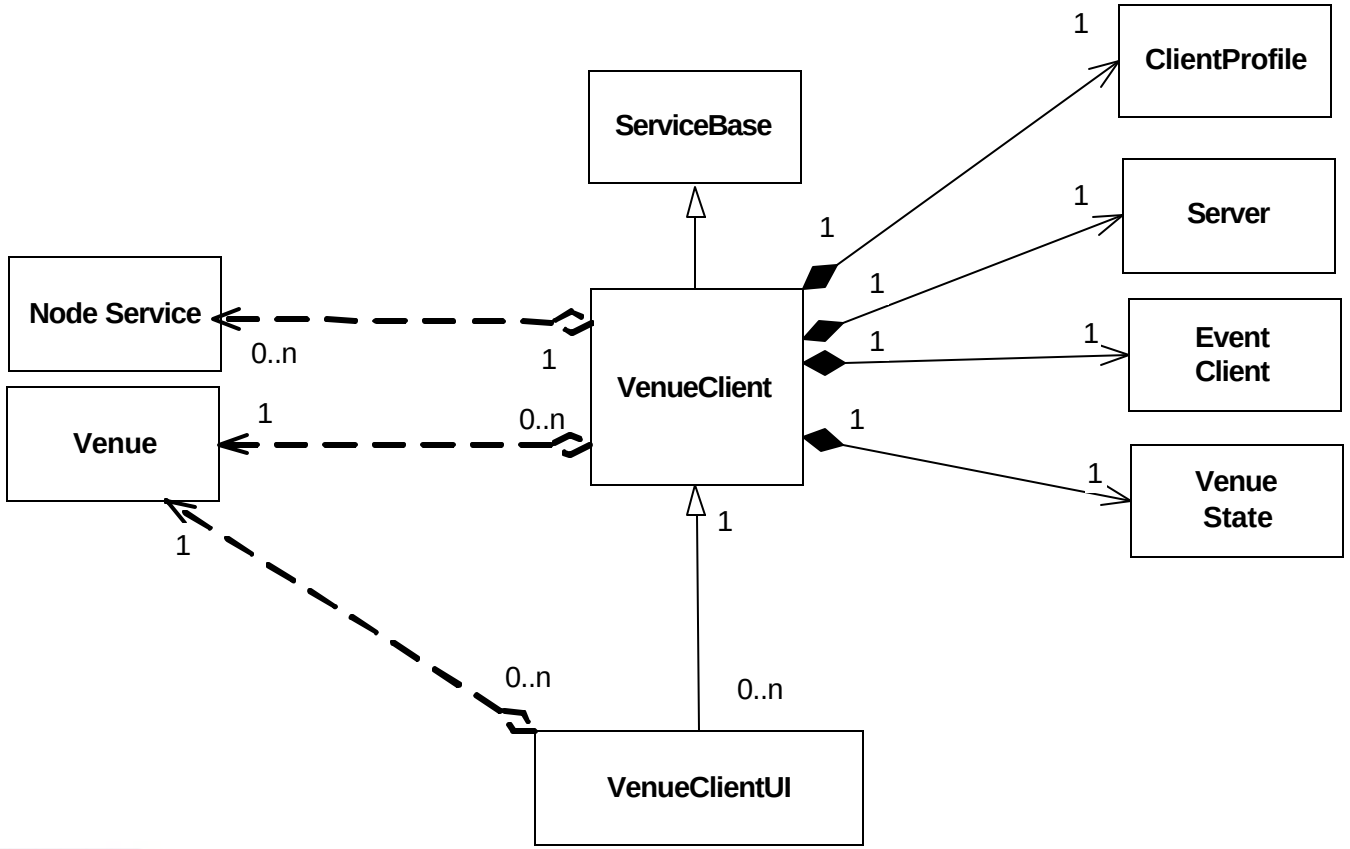
Virtual Venues



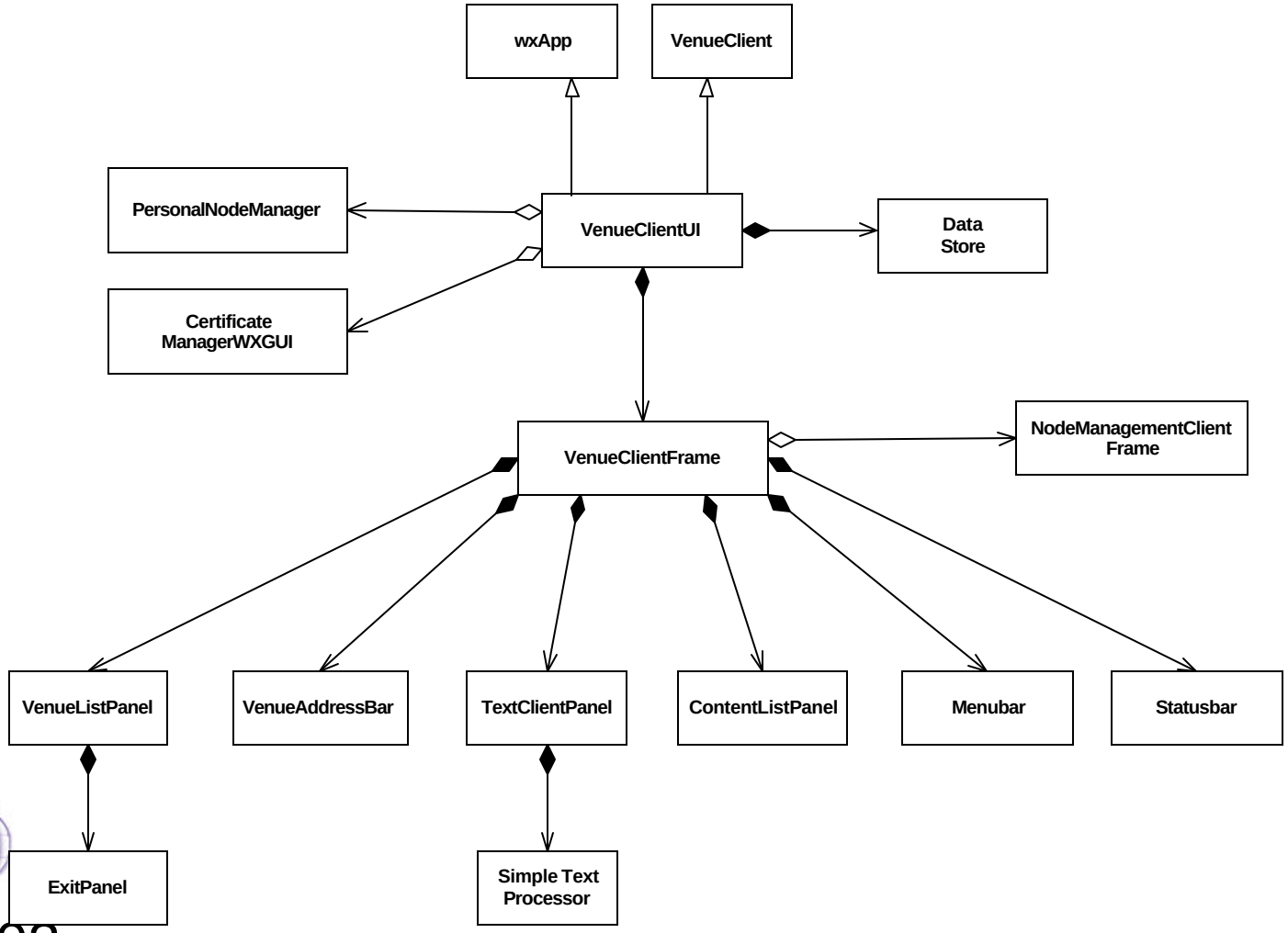
ACCESS GRID

Retreat 2003

Virtual Venues Client



Virtual Venues Client II



Network Services

- Network Services expose
  - Capabilities
  - Data Interfaces
  - Control Interfaces
  - Configuration Client Interface
- These provide enough information for both manual and programmatic use of Network services

Node Services

- Node Services are Web Services that provide capabilities or expose resources
- Current Services are built using a simple python class that encapsulates details of the actual software
  - AGService.py
- Soon Node Service will be able to be language independent

Application Services

- Application Services exposed in the Virtual Venues through an Adapter
- On the Venue Side the service looks *almost* like a Venue Client
- On the Application Side the venue looks like an interactive Administrative user
- Users connect to the Service through external (to the Access Grid) network interfaces

Developer Entry Points

- Developers wanting to add effort to the development of the AG could contribute to various aspects:
  - Core development
  - Shared Applications
  - Network Services
  - Node Services

Core Development

- Any Developer is welcome to help complete the Core Toolkit
- We'd like to see proposals to augment the Core Toolkit, so we can work with the developers to come up with a design that satisfies everyone.
- Then developers can implement the design.

Network Services

- Anyone can (and should!) comment on the Network Service Architecture Draft
- Once the Network Service Design Draft is complete it will need external review
- Anyone can develop network services

Node Services

- Node Services provide rich collaboration
- New Node Services could be developed for:
  - Higher Quality Video
  - Higher Quality Audio
  - Layered Media
  - Device Control
    - Gentner
    - Cameras

Shared Applications

- Example Applications have been built for:
  - A Shared Browser
  - A Shared Presentation Viewer
  - A Shared Image Viewer
- Many other shared applications are needed.
- These are simple to develop and add instant, significant value.

Plans for the Next Year

- Documentation Available to interested Developers
- AGTk 2.0 released with core functionality robust
- Iterate:
  - Propose extensions
  - Design extension
  - Implement extensions

Define a formal process?